Exhibit 42

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION,

Petitioner

v.

ACQIS LLC,

Patent Owner

Case No. IPR2014-01469
Patent: RE42,814

PATENT OWNER'S RESPONSE

Mail Stop **PATENT BOARD**
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

LIST OF EXHIBITS

Exhibit 2007	Horst, TNet: A Reliable System Area Network, 15 IEEE Micro 37 (Feb. 1995) ("Horst") as annotated by Bruce Young.
Exhibit 2008	Horst, TNet: A Reliable System Area Network, 15 IEEE Micro 37 (Feb. 1995) ("Horst") as annotated by Bruce Young.
Exhibit 2009	Bogaerts, Application of the Scalable Coherent Interface to Data Acquisition at LHC (Oct. 1996) as annotated by Bruce Young.
Exhibit 2010	Bogaerts, Application of the Scalable Coherent Interface to Data Acquisition at LHC (Oct. 1996) as annotated by Bruce Young.
Exhibit 2011	Declaration of Spencer Scott dated April 8, 2015.
Exhibit 2012	Computer Architecture, A Quantitative Approach, J. Hennesy & D. Patterson, 1990 (Excerpts)
Exhibit 2013	Computer Organization and Design, the Hardware/Software Interface, D. Patterson & J. Hennessy, 1998 (excerpts)
Exhibit 2014	PCI Express System Architecture, by R. Budruk et al, 2004 (excerpts)
Exhibit 2015	PCI and PCI-X Hardware and Software Architecture & Design, E. Solari & G. Willse, 2001 (excerpts)
Exhibit 2016	Intel Nontransparent Bridge Spec., 21554 PCI-to-PCI Bridge, Dec. 1996
Exhibit 2017	VITA Journal, Multiprocessing on PCI and CompactPCI, "Here Today – Multiprocessing on PCI and CompactPCI", October/November/December 1998
Exhibit 2018	US Patent No. RE 42,814
Exhibit 2019	PCI Express Base Specification, Revision 3.0, November 10, 2010
Exhibit 2020	RapidIO Technology Comparisons _ PCIe and Ethernet vs. RapidIO
Exhibit 2021	Declaration of Volker Lindenstruth, dated June 11, 2015
Exhibit 2022	Deposition of Bruce Young, dated May 20, 2015, Case No. IPR2015-01462 & 2014-01469
Exhibit 2023	Deposition of Bruce Young, dated May 21, 2015, Case No. IPR2015-01462 & 2014-01469
Exhibit 2024	IBM Dictionary of Computing, G. McDaniel, 1994 (excerpts)
Exhibit 2025	Deposition of Spencer Scott, dated June 1, 2015, Case No. IPR2015-01462 & 2014-01469
Exhibit 2026	Declaration of Christopher Butler, dated March 30, 2015

I. Introduction

At first glance, there is superficial similarity between the two primary references presented to the Board and the '814 claims at issue here. But that similarity belies critical, patentable distinctions between the '814 inventions and the Horst and Bogaerts references. As explained below and confirmed by Dr. Lindenstruth, even though a PCI transaction can eventually be completed in Horst and Bogaerts, these transactions occur on a remote parallel PCI bus and are never serialized as required by the claims at issue. Ex. 2021 at ¶¶ 138, 141, 156. This distinction, and others, differentiates the '814 claims from the prior art in the instituted grounds.

The two primary references applied by the Board solve very different problems than the '814 inventions. Horst and Bogaerts are designed for massively-parallel systems in which one CPU can share the memory space of another CPU. Sharing CPU memory space between multiple CPUs is not supported by the Peripheral Component Interconnect ("PCI") local bus standard because of address collisions and overlapping address spaces. Ex. 2016 at 6; Ex. 2017 at 3; Ex. 2020 at 3.

Each CPU in these parallel systems has the same processor address spaces. Ex. 2021 at 94, 108. The PCI standard's addressing scheme is only designed to work with a single host processor's address space. Ex. 2016 at 6; Ex. 2017 at 3;

Ex. 2020 at 3. A read-or-write PCI transaction using a PCI address would cause a collision because every CPU memory would have the same address. Ex. 2016 at 6; Ex. 2017 at 3; Ex. 2020 at 3; Ex. 2021. To solve this addressing problem, Horst and Bogaerts discarded the PCI standard bus protocol and introduced new protocols—TNet and SCI. These protocols permit one CPU to directly address the CPU memory of another CPU. Ex. 2021 at ¶¶ 109, 133, 155. These protocols also permit generation of a PCI transaction at a remote device. But notably, these protocols eliminate the PCI transaction on the CPU side of the network. Eliminating the PCI transaction on the CPU side was necessary to enable the massively-parallel systems. Because PCI was in widespread use for peripheral devices, Horst and Bogaerts needed to provide access to these devices. To maintain PCI compatibility, Horst and Bogaerts implemented hardware and software interfaces *on the peripheral side* designed to generate a PCI transaction for transmission only over the remote, parallel PCI bus. Neither reference employs PCI transactions within its own TNet or SCI fabric, and neither ever puts a PCI transaction on a serial bus.

The '814 invention solves a very different problem than Horst and Bogaerts were addressing. The '814 invention is directed at increasing communication speeds between the computer and peripherals. The inventor recognized that, as CPU speeds increased, communications with peripheral devices would need to keep up. But the problem with keeping up was a limiting industry standard, PCI. PCI had been widely adopted and could not be replaced without requiring peripheral manufacturers to replace drivers (the software that controls the peripheral connection). Ex. 2012 at 20; Ex. 2013 at 6; Ex. 2021 at ¶¶ 79, 97.

Realizing that the industry had invested heavily in PCI and would not easily abandon it, even if it was too slow, the '814 inventor developed a system to speed up PCI transactions that was completely compatible with existing peripheral devices and drivers. As described in the specification and explained further below, one key to the invention was to serialize the otherwise parallel PCI bus transactions to increase communication speeds for peripherals. Another key was to maintain the PCI transaction address bits as generated on the CPU side of the network so that the serialized communications were compatible with existing peripheral devices and their drivers. Ex. 2020 at 4:50-58. The claims reflect this invention with their focus on the serialized PCI bus transaction, the serialized address bits of the PCI bus transaction, and the specific architecture in which the serialized PCI bus transaction is transmitted from the processor-bridge to the peripheral. The inventor chose claim terms like "directly coupled," "extending from," "without an intervening Peripheral Component Interconnect (PCI) bus," and "address and data bits of PCI bus transaction in serial form" to distinguish the claimed architecture from other systems. Ex. 1001, claims 24 & 31.

Because Horst and Bogaerts solve a very different problem, it is not surprising that they only match up superficially with the '814 claims. But, as the arguments below describe in detail, when the specific language of the claim limitations are considered, such as "directly coupled," "extending from," "without an intervening Peripheral Component Interconnect (PCI) bus," and "address and data bits of PCI bus transaction in serial form," the references fail to disclose the claimed invention, and the secondary references do not cure the failure. Specifically, to implement their parallel-processing systems with shared CPU memories, Horst and Bogaerts deliberately eliminated the very principle that the '814 claims address—a CPU-side PCI bus transaction that increases communication speed. Ex. 1009 at 2, Fig. 2; Ex. 1011 at 16, Fig. 13. And these references never used a serialized PCI bus transaction, only parallel. Ex. 2021 at ¶¶ 138, 141, 156.

Figure A[1] illustrates the difference between Horst and the traditional PCI-based system. The traditional system is illustrated in Figure 1-2 from the PCI specification and is shown on the left. Figure 2 of Horst is shown on the right. The traditional system included a CPU side (labeled as "A") that communicated

[1] To avoid confusion with the numbered figures excerpted from the cited references, figures in this Response are lettered (e.g., Figure A).

with a peripheral side (labeled as "B") through a parallel PCI local bus (labeled as "C"). Horst removed the PCI bus and replaced it with a TNet network and a proprietary protocol. (Bogaerts similarly removed the PCI bus and replaced it with SCI. Ex. 1011 at Figure 15.) The TNet system (shown on the right) has a CPU side (labeled as "A_1") that communicates with a peripheral side (labeled as "B_1") through the TNet link (labeled as "C_1"). Figure A illustrates that in the upper portion of Horst, which is the CPU side, there is no PCI bus. And, consequently, the TNet link does not generate PCI bus transactions or PCI transactions at the CPU side.

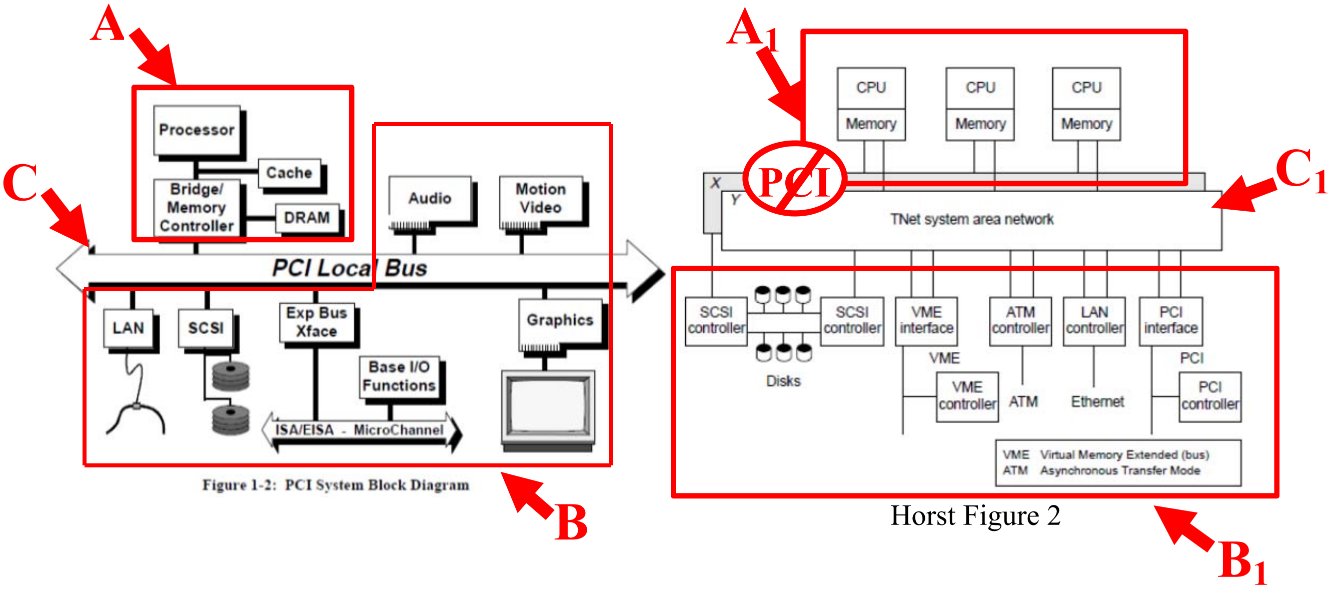


Figure 1-2: PCI System Block Diagram

Horst Figure 2

Figure A: Comparison of the traditional PCI-based system and TNet

Bogaerts similarly removes the PCI bus. Ex. 1011 at Figure 13.

Just because a PCI transaction is eventually completed elsewhere in Horst and Bogaerts on a remote PCI bus—on the peripheral side—does not make Horst or Bogaerts the same as the '814 invention. The '814 patent claims a very specific architecture linked to serializing a PCI bus transaction on the CPU side of a network as opposed to the peripheral side. Ex. 1001, claims 24 & 31. Bogaerts and Horst did not and could not implement this CPU-side PCI serialization and still achieve their massively-parallel processor goals. Accordingly, neither Horst nor Bogaerts teach the limitations recited in the claims under review. And no other references fill, or are even offered to fill, the gaps.

Further, Bogaerts is not even prior art. Petitioner points to a submission date and an undated library stamp as proof of a publicly-available date. But the library stamp is not associated with any date—and is definitely not associated with the submission date. *Compare* Ex. 1011 at 1 (library stamp) *with* Ex. 2026 at 12 (date, but no library stamp). And Petitioner chose not to investigate the library stamp with the library to confirm a date. Further, evidence from Dr. Lindenstruth, one of the co-authors, establishes that the only date printed on the face of Bogaerts was merely a submission date to the CERN committee to which the project team reported. This date does not indicate public availability. In fact, Dr. Lindenstruth thought it was unlikely that the Bogaerts reference would have been publicly available on this date due to confidentiality obligations. Ex. 2021 at ¶ 170. Thus, Petitioner failed to meet its burden to show Bogaerts was even prior art.

II. Background and Summary of Arguments

A. Related Matters

IPR2014-01462 has been instituted on related U.S. Patent No. 8,041,873.

The '814 patent has been asserted in two pending district court litigations. *ACQIS v. Alcatel-Lucent USA Inc.*, 6:13-cv-638, pending before the United States District Court for the Eastern District of Texas and *ACQIS, LLC v. EMC Corporation*, 1:14-cv-13560, pending and stayed before the United States District Court for the District of Massachusetts.

B. Grounds Upon Which Trial is Instituted

Ground 1: Claims 24 and 31-33 are obvious over Horst (Ex. 1009), Mathers (Ex. 1014), and the LVDS Owner's Manual (Ex. 1019);

Ground 2: Claims 31-33 are obvious over Bogaerts (Ex. 1011), Gulick (Ex. 1010), Mathers (Ex. 1014), and James (Ex. 1018).

C. Background of the Invention of the '814 Patent

The '814 patent describes many features that could be used in modularized computer systems. At its most basic level, the patent describes a computer system that includes a removable computer module and the enclosure into which the module is inserted. But the patent goes much deeper into different modular architectures that are (1) compatible with existing communication protocols and (2) designed for increasing communication speed with peripherals in a modular computer system. Ex. 2013 at 9; Ex. 2018 at 3:30-67; 4:50-58; Ex. 2021 at ¶¶ 79-80. The '814 claims are directed at an industry-compatible architecture that speeds up communications between the attached computer module ("ACM") and peripheral devices. Ex. 2012 at 20; Ex. 2013 at 9; Ex. 2018 at 4:50-58. More specifically, the '814 claims are directed to a CPU-side (as opposed to a peripheral-side) architecture that speeds up PCI bus transactions with peripheral devices by serializing the PCI bus transaction, including serializing the PCI transaction address. Ex. 1001 at Fig. 18; Ex. 2018 at 3:32-67; 4:50-58.

One significant advantage to the use of PCI as the communication protocol between the CPU side and the peripheral side is speed of communication with local peripheral devices. Ex. 2001 at 17-21. The invention of the '814 patent retains that advantage, as well as compatibility with existing devices and PCI drivers, while boosting speeds further by making the PCI communication serial rather than parallel. Ex. 2018 at 3:32-67; 4:50-58.

Note that PCI is an industry standard. Ex. 2001 at 22. Standards are very important in computer architecture. Standards ensure that multiple designers can develop hardware and software and that it will all work together. Even minor deviations from a standard will result in incompatible hardware and software. Ex. 2012 at 20; Ex. 2013 at 9; Ex. 2022 at 10:20-25; 11:10-18. Therefore, adherence to the specification of a standard (such as the PCI standard local bus specification) is critical for interoperability with other components of a system. The '814 adheres to the PCI standard. Ex. 1001 at cl. 24, 31; Ex. 2018 at 4:50-58. SCI and TNet do not. Ex. 1009 at 2; Ex. 1011 at 16, Fig. 13.

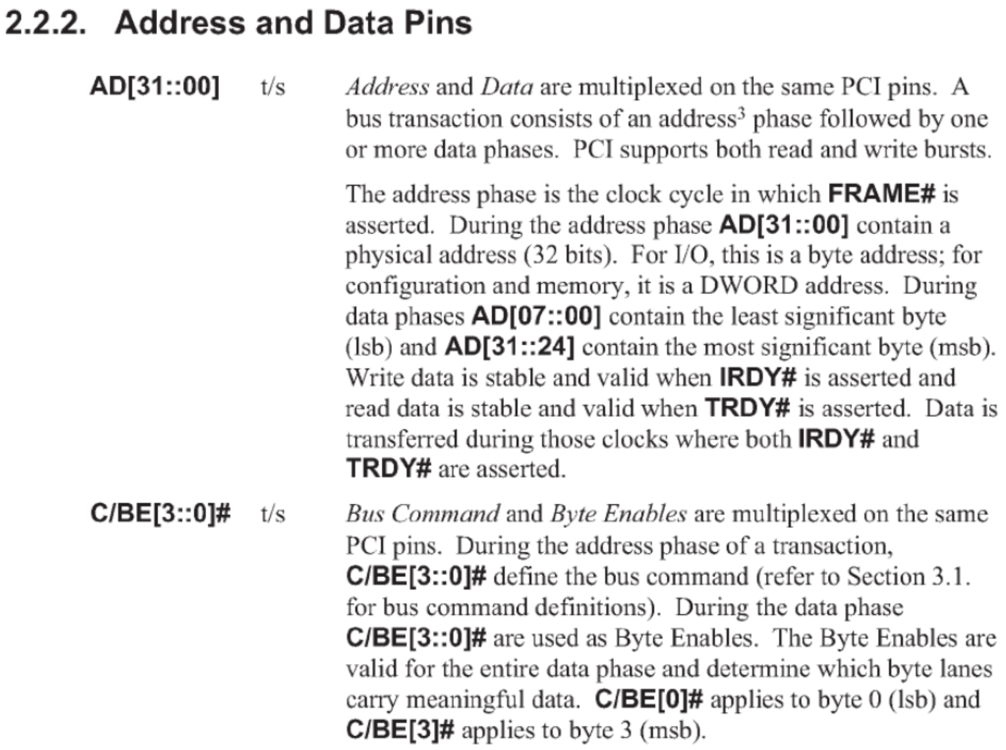
1. The PCI Standard

In the late 1990s, the PCI bus was ubiquitous in computers and provided interconnection between processors and peripheral devices. Ex. 2013 at 9; Ex. 2021 at ¶¶ 59-60. This bus operated according to the PCI standard, which was developed by Intel and maintained by an industry group, the PCI Special Interest Group ("PCI-SIG"). Ex. 2001 at 17-22; Ex. 2021 at ¶ 59. Peripheral manufacturers were expected to provide drivers for their devices so that they could interact with the PCI bus found on nearly every computer. Ex. 2001 at 227; Ex. 2014 at 3. And the PCI standard provided the instructions for designing those drivers. E.g., Ex. 2001 at 227.

The PCI bus was a 32-bit wide parallel bus with multiplexed address and data bits and separate command and timing control lines. Ex. 2001 at 17, 23. According to the published PCI standard, a PCI bus transaction consists of an address phase followed by one or more data phases. Ex. 2001 at 25. Each device on a PCI bus must have a unique physical address. Ex. 2014 at 6-7; Ex. 2015 at 9, 12; Ex. 2016 at 6. And, any device on the PCI bus can read or write data directly to or from another device's address space. Ex. 2015 and 6-9, 12; Ex. 2021 at ¶ 63.

The PCI specification defines specific bus transactions that use the three different PCI physical address spaces, including: memory read-and-write transactions, I/O read-and-write transactions, and configuration read-and-write transactions. Ex. 2001 at 25, 42; Ex. 2014 at 3, 6-7, Ex. 2015 at 9. These transaction types are defined by bus commands that tell the PCI devices which address space is being utilized in the particular transaction. Ex. 2001 at 25, 37; Ex. 2015 at 9. Specifically, the PCI transaction types define different address spaces and different address formats for addressing PCI devices on a PCI bus. Ex. 2001 at 25.

A PCI standard bus transaction requires the PCI bus command specific to the transaction type and a corresponding physical PCI address in the correct format and to the correct address space. Ex. 2001 at 25, 37, 42, 45 ("Byte lane swapping is not done in PCI"); Ex. 2021 at ¶ 73. The PCI bus transaction must comply with a particular format for peripheral devices and their PCI drivers to understand it. Figure B illustrates the specific requirements of PCI address and data pins from the PCI standard local bus specification. Ex. 2001 at 25.



2.2.2. Address and Data Pins

AD[31::00] t/s *Address* and *Data* are multiplexed on the same PCI pins. A bus transaction consists of an address[3] phase followed by one or more data phases. PCI supports both read and write bursts.

The address phase is the clock cycle in which **FRAME#** is asserted. During the address phase **AD[31::00]** contain a physical address (32 bits). For I/O, this is a byte address; for configuration and memory, it is a DWORD address. During data phases **AD[07::00]** contain the least significant byte (lsb) and **AD[31::24]** contain the most significant byte (msb). Write data is stable and valid when **IRDY#** is asserted and read data is stable and valid when **TRDY#** is asserted. Data is transferred during those clocks where both **IRDY#** and **TRDY#** are asserted.

C/BE[3::0]# t/s *Bus Command* and *Byte Enables* are multiplexed on the same PCI pins. During the address phase of a transaction, **C/BE[3::0]#** define the bus command (refer to Section 3.1. for bus command definitions). During the data phase **C/BE[3::0]#** are used as Byte Enables. The Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data. **C/BE[0]#** applies to byte 0 (lsb) and **C/BE[3]#** applies to byte 3 (msb).

Figure B: PCI Local Bus Standard Specification – Address and Data Pins

2. The '814 claimed invention increases communication speed while maintaining the CPU-side PCI bus transaction

PCI was the industry standard in the late 1990s. Ex. 2013 at 6. Executing a PCI bus transaction was a requirement for any new, commercially-acceptable computer system. Ex. 2012 at 20; Ex. 2013 at 6; Ex. 2021 at ¶ 47. But PCI had a major problem—speed constraints. The PCI standard specified 32-bit parallel communications. Ex. 2001 at 17. The inventor of the '814 patent recognized that parallel communications would eventually become a speed bottleneck for communicating with peripheral devices. Ex. 2018 at 4:50-58; Ex. 2021 at ¶¶ 44, 55.

The inventor recognized that a serial communication scheme would allow fast, reliable communication. Ex. 2018 at 3:32-67, 4:58. Serial communication is faster than parallel communication because parallel communications are slowed by data skew and electromagnetic interference between closely-positioned parallel lines (known as cross-talk). Ex. 2021 at ¶¶ 44, 55, 58. But designing a new, serial communication protocol was not practical. The trick was to use the existing PCI standard and make it faster for communicating with peripheral devices. And the solution is in the '814 claims—a new architecture that serializes the PCI bus transaction from the CPU side of the network (rather than using an entirely new standard and moving the old parallel PCI bus to the peripheral side). By retaining PCI communication on the CPU side, the system does not require new drivers or peripheral devices. Ex. 2018 at 4:40-58; *also* Ex. 2014 at 3. Further, the claimed invention does not suffer performance penalties because the serialized PCI connection is as fast as the PCI standard and no additional address translations or software intervention is required. *Id.*

By using the industry standard PCI bus transaction, this new, claimed architecture not only increased communication speeds, but it was also compatible with existing peripherals and their drivers. Ex. 2014 at 3; Ex. 2018 at 4:50-58. Further, by maintaining the PCI transaction, the new architecture also took advantage of the plug-and-play capabilities of the PCI standard. Ex. 2014 at 3;

Ex. 2018 at 4:50-58. Accordingly, the '814 patent describes—and claims—the solution to the problem caused by the PCI standard's use of parallel communications while maintaining compatibility with that standard.

3. The '814 claims specifically claim the improved architecture that maintains the CPU-side PCI bus transaction

The claims specifically address the architecture required to implement PCI bus transactions on the CPU side and to achieve the speed gains.

Referring first to the speed gains, the claims address the speed gains with the use of the low voltage differential signal ("LVDS") channels. Ex. 1001 at cls. 24, 31. These channels communicate data serially. *Id.* The claim language in both claim 24 and 31 make clear that these channels communicated data in serial form. For example, claim 24 refers to "two unidirectional serial channels" and "in serial form." Ex. 1001, claim 24. This serial form of a PCI bus transaction increases speed over parallel forms of a PCI bus transaction.

Referring now to the CPU-side architecture, the claims define a specific architecture to distinguish it from the peripheral-side architecture. Ex. 1001 at cls. 24, 31, FIG. 18. For example, claim 24 recites a north bridge "directly coupled" to the microprocessor. Ex. 1001 at cl. 24. This "directly coupled" limitation establishes the north bridge as part of the CPU side and not the peripheral side of the system. Ex. 1001 at cl. 24; Ex. 2022 at 17:15-19:25; 85:2-8. The claim also requires that the north bridge convey "address and data bits of the PCI bus transaction in serial form." Accordingly, the PCI bus transaction—including the address and data bits—exist on the CPU side of the system as opposed to only on the peripheral side.

Claim 24 specifically requires that the north bridge communicate the serialized PCI bus transaction data bits over a low voltage differential signal ("LVDS") serial channel that extends from the north bridge. Ex. 1001, cl. 24; Ex. 2022 at 85:2-8. Mr. Young, Petitioner's expert, explained that "extends" means that the LVDS lines are "directly coupled to the north bridge and go somewhere else." Ex. 2022 at 44:6-22. The inventor selected the "extends" language to limit claim 24 to a CPU-side method for serializing PCI bus transactions.

Claim 31 also requires CPU-side, rather than peripheral-side, architecture. For example, claim 31 recites that the "peripheral bridge" is coupled to the microprocessor "without any intervening Peripheral Component Interconnect (PCI) bus." Ex. 1001 at cl. 31. This language is similar in concept to claim 24's "directly coupled."

D. Background of Horst and Bogaerts

Horst and Bogaerts did not address the communication speeds between the computer and its peripheral devices. These references were directed at protocols for massively-parallel systems that needed to share CPU memory among multiple processors. While both protocols enable communications between a computer and a peripheral device, they do so by eliminating the CPU-side PCI transaction and replacing it with a brand-new communication protocol. They both require that a PCI transaction be created on the peripheral side of the network by a custom piece of hardware and software. Simply put, Horst and Bogaerts do not solve the peripheral communication speed problem because they were not trying to.

1. Horst did not solve the communication-speed problem and does not use a CPU-side PCI bus transaction

Horst (TNet) interconnected hundreds of processors to share memory among the individual CPUs while allowing the processors to share I/O resources. Ex. 1009 at 1. The author evaluated existing communications standards, which included PCI, but "found nothing that met all the requirements." Ex. 1009 at 2. "This forced [the author] into the difficult job of designing a completely new network." Ex. 1009 at 2. As a result, the author specifically chose to avoid the PCI standard bus transaction protocol on the CPU side as well as any other existing protocol in favor of a "completely new network." *See* Ex. 1009 at 2, 5, Fig. 5.

The "completely new network" described by Horst requires a specially-designed intelligent processor and I/O nodes driven by application-specific integrated circuits (ASICs) to generate and transmit TNet transactions. Ex. 1009 at 6-7, Figs. 7, 8. The "completely new network" described by Horst uses virtual (rather than physical) addressing and address translation and purposefully avoids memory-mapped I/O and physical addressing (e.g., PCI). Ex. 1009 at 2, 6. Stated differently, Horst needed to introduce new peripheral-side hardware and software to enable a PCI transaction because his new protocol eliminated PCI transactions from the CPU side. The result is a system riddled with proprietary hardware and software interfaces to facilitate communication between the CPU and legacy devices. And all communications on the CPU side are done exclusively in the proprietary TNet protocol. Ex. 1009 at 2, 4-6, 7 Figs. 2, 5, 7, 8, 9. Figure C, excerpted and annotated from Horst figure 2, illustrates the CPU side and peripheral side of the TNet system. Note that the PCI communication exists *only* in the peripheral side.

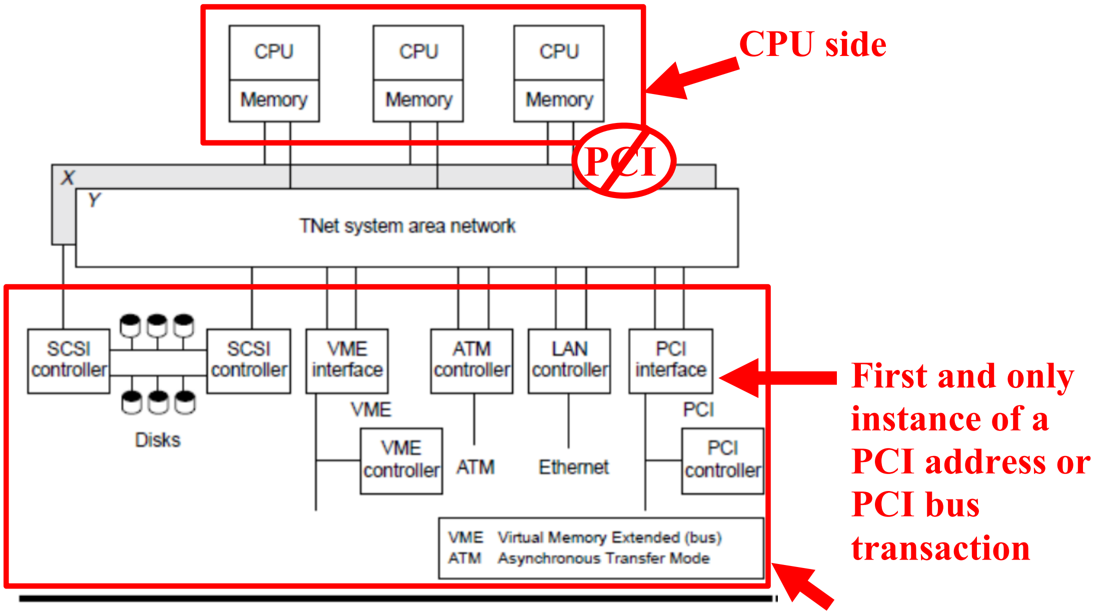


Figure 2. CPU cluster connected with the TNet system area network.

Figure C: Annotated TNet

While there is a lack of detail around the exact routing mechanisms and address schemes used for the TNet protocol, as Dr. Lindenstruth explains, it is clear from the TNet reference that there is no PCI standard bus transaction on the TNet link. Ex. 2021 at ¶¶ 141-43; also Ex. 1009 at 2, 4-6; Ex. 2001 at 37-39, 42, 47; Ex. 2015 at 12; Ex. 2016 at 6. The disclosed TNet transaction types are only read, write, and unacknowledged write. Ex. 1009 at 5, Fig. 5. The TNet packets and transaction types do not allow for the PCI standard bus transactions, including memory read and write, I/O read and write, and configuration read and write. Ex. 1009 at 4-5, Fig. 5, Ex. 2001 at 37-39; Ex. 2021 at ¶ 105. Configuration transactions are required by the PCI standard in order to initialize the PCI bus to accept PCI transactions. Ex. 2001 at 39, 42, ; Ex. 2021 at ¶ 95. Because configuration transactions are not supported on TNet, TNet *cannot* communicate PCI bus transactions. Even if the PCI addressing scheme could work over TNet, which it cannot, there is no way to configure the TNet devices to operate with PCI transactions. *Id.* at ¶¶ 141-42, 152.

Nor does Horst disclose any mechanism for the TNet packets to communicate PCI bus commands, which would indicate a type of PCI standard bus transaction. Ex. 1009 at 4-5; Ex. 2001 at 21; Ex. 2021 at ¶ 105. Instead, the only commands Horst discloses for TNet relate to link-level flow control, initialization, and error signaling. Ex. 1009 at 3, Table 1. In summary, TNet does not support CPU-side PCI bus transactions.

Further, the TNet packet as disclosed in Horst does not disclose PCI standard addresses. Ex. 1009 at 5, Fig. 5. First, TNet discloses that it does not support memory-mapped I/O (Ex. 1009 at 6), which is the addressing scheme used in PCI memory read-and-write transactions. Ex. 2015 at 9-10, 12. Second, TNet does not support PCI standard addresses. Ex. 2001 at 42; Ex. 2014 at 6-7. TNet addresses are virtual, and PCI addresses are physical, which are two vastly different addressing schemes. Compare Ex. 1009 at 2, 6 with Ex. 2001 at 42; also Ex. 2012 at 3-10; Ex. 2013 at 6; Ex. 2021 at ¶¶ 37-40. Only after a TNet packet is routed to a TNet node connected to a PCI bus does the intelligent ASIC in the PCI bus node generate a PCI bus transaction based on the TNet packet information and other information contained in the ASIC. Ex. 1009 at 6-7, Figs. 7, 8; Ex. 2021 at ¶¶ 94-96; Appendix A.

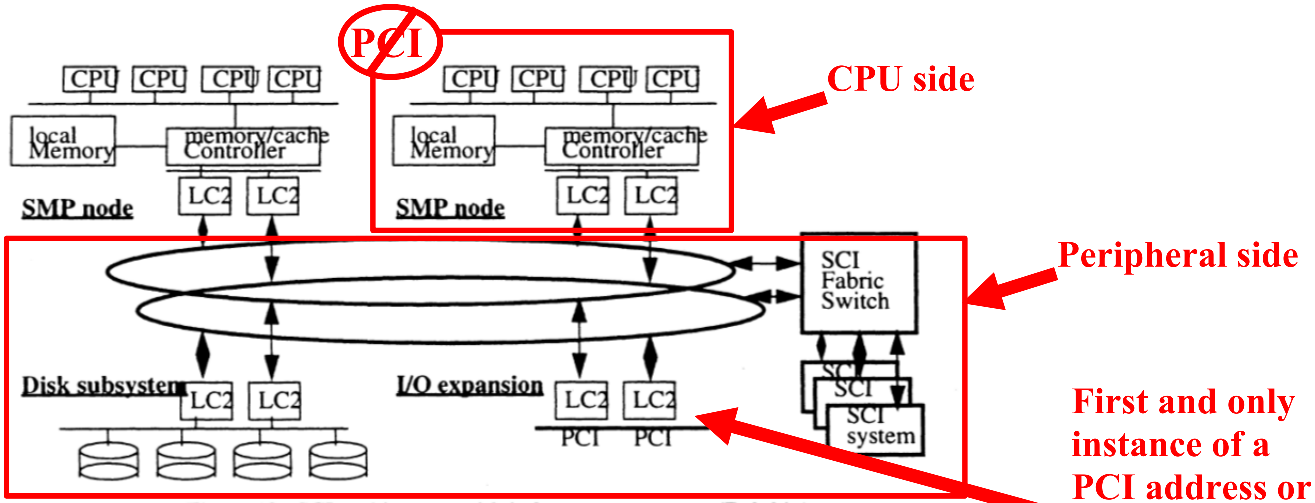
2. Bogaerts did not solve the communication-speed problem and does not use a CPU-side PCI bus transaction

Bogaerts describes another massively-parallel computer system that rejected CPU-side PCI in favor of the SCI protocol. The goal, similar to the goal of TNet, was to establish transparent access to memory attached to other CPUs. Bogaerts discloses a system intended to be a scalable data acquisition system for the large hadron collider (LHC) program at the European Organization for Nuclear Research (also known as CERN). Ex. 1011 at 2-6; Ex. 2021 at ¶ 108. The project, referred to as RD24, utilized a special communication protocol called the Scalable Coherent Interface ("SCI"). Ex. 1011 2-6.

SCI is a serial communication channel that provides computer-bus-like services. Ex. 2021 at ¶ 108. SCI supports distributed, shared memory with optional cache coherence and message passing. Ex. 1011 at 4-6; Ex. 2021 at ¶ 108. The SCI specification does not support the 32-bit PCI address spaces or PCI bus transactions. *Id.* at ¶ 109. Instead, SCI uses a single 64-bit address space and defines its own transaction types. *Id.*

Because the addressing scheme is different between PCI and SCI, adapters are required to translate the addresses into a suitable format. E.g., Ex. 1011 at 8-10; Ex. 2021 at ¶¶ 108-110. In the RD24 project, the PCI-SCI adapters discarded the PCI bus commands and translated the PCI address from the initiating PCI bus, typically in 32-bit format, into SCI's 64-bit address space. The PCI-SCI adapters also discarded the PCI bus commands, which were not sent over the SCI ring. *Id.* Instead, the appropriate SCI transaction type was selected by the intelligent SCI adapter. *Id.* at ¶ 110. Similarly, on the other end, the reverse occurred. *Id.* Notably, the PCI address from the initiating bus was not the same PCI address generated by the PCI-SCI adapter on the receiving bus. *Id.* In other words, no PCI information was transmitted over the SCI bus.

Additionally, a PCI-SCI adapter was not intended to be coupled directly to the CPU. Bogaerts discloses a PRO-SCI adapter for that purpose. The PRO-SCI adapter couples the CPU to an SCI ring. The PRO-SCI adapters translated the address from the transaction coming off the CPU-memory bus into SCI's 64-bit address space. Ex. 2021 at ¶ 165. Because the PRO-SCI adapter takes in a CPU-memory bus transaction, there is never a PCI bus transaction on the CPU side to be translated into SCI in the first place. *Id.* Figure D, below, is annotated figure 15 from Bogaerts. Figure D shows a system interconnected using PRO-SCI adapters and PCI-SCI adapters.



CPU side

Peripheral side

First and only instance of a PCI address or PCI bus transaction

Figure 13: SCI architecture with industry components (Dolphin)

Figure D: Annotated Bogaerts Architecture

Importantly, as described in detail above, neither the PCI-SCI adapters nor the PRO-SCI adapters transmit a PCI address over the SCI ring. *Id.* at ¶ 166.

III. CONSTRUCTION OF AND SUPPORT FOR THE INSTITUTED CLAIMS

A. The Standard for Claim Construction in These Proceedings

The claims are to be given their broadest reasonable interpretation in light of the specification. *See* 37 C.F.R. § 42.100(b).

B. Patent Owner's Proposed Claim Constructions

Patent Owner's proposed claim constructions for selected claim terms and clauses are provided below. Any claim terms or clauses not expressly construed herein should be given their plain and ordinary meaning as understood by one skilled in the art.

1. Peripheral Component Interconnect (PCI) bus transaction

The proper construction for "Peripheral Component Interconnect (PCI) bus transaction" and "PCI bus transaction" is the one the Board adopted on institution—"Peripheral Component Interconnect (PCI) industry standard bus transaction." Any other reading of the term would render the phrase "PCI" meaningless.

The person of ordinary skill in the art ("POSA")[2] would understand that a "PCI bus transaction" refers to a **PCI standard bus transaction** and not just any type of bus transaction. The PCI industry standard was well known at the time of the invention, and the POSA would not confuse the PCI transaction for any other transaction. Ex. 2021 at ¶ 114. The PCI version 2.1 local standard bus specification is dated June 1, 1995, three years prior to the '814 patent priority date. Ex. 2001 at 1.

Therefore, the Board's construction is the proper construction under BRI.

2. Extending from

The proper construction for "extending from" as used in claim 24 ("said second LVDS channel extending from said north bridge to convey said address and data bits of PCI bus transaction in serial form") is "directly coupled to." This meaning is consistent with the '814 patent specification and claim language, as well as Petitioner's expert's interpretation of the '814 Patent. For example, Fig. 18 of the '814 Patent shows the second LVDS channel "extending from" or "directly coupled to" the North Bridge 1805 as described in claim 24. Petitioner's expert also interprets "extending" as "directly coupled to." For example, during his deposition, Mr. Young described "extending" as used in claim 24 as "directly

[2] For a definition of the POSA in this case, please see Dr. Lindenstruth's declaration.

coupled to the North Bridge, those are lines that connect North Bridge [sic] and go somewhere else . . . '[e]xtending to me implies they're directly connected. It's not somehow coupled to it. It's extending from the pins of that circuit . . . I think one of ordinary skill would interpret it that way." Ex. 2026 at 44:6-22.

Therefore, under BRI, the proper construction of "extending from" is "directly coupled to."

IV. THE CLAIMS ARE PATENTABLE OVER THE INSTITUTED GROUNDS

Horst and Bogaerts do not disclose key limitations of claims 24, 31, 32, or 33. These claims are very specific on what is being communicated, how the communication happens, and where the communication happens. And these specifics distinguish the claimed invention from Horst and Bogaerts. Combining these references with the secondary references does not cure the problem. Notably, these references are not even offered to teach the limitations missing from Horst and the Board did not institute review on these secondary references for Bogaerts.

A. The Instituted Claims Are Patentable Under 35 U.S.C. §§ 102 and 103

1. Claim 24 is not rendered obvious over Horst, Mathers, and the LVDS Owner's Manual (Ground 1)

Claim 24 requires a "north bridge to communicate address and data bits of PCI bus transaction in serial form." The claim also requires that the north bridge be "directly coupled" to the microprocessor unit. Horst does not teach this architecture. Instead, as discussed further below, Horst uses only TNet transactions on the CPU-side of the bus. Ex. 2021 at ¶¶ 94, 101, 129-31. Horst creates a parallel (not serial) PCI bus transaction on the peripheral side of the computer system using specialized hardware and software. Horst never serializes any PCI bus transaction, even the peripheral-side PCI bus transaction. *Id.* ¶ 137-38. The only PCI bus transaction shown in Horst is parallel and is on the peripheral side. Thus, Horst fails to teach this claim limitation. And no other reference is put forward to teach this portion of the claimed invention.

Claim 24 further requires a "second LVDS channel extending from said north bridge to convey said address and data bits of PCI bus transaction." Again, Horst does not teach this architecture. Horst does use a serial communication method between the CPU side of the system and the peripheral devices, but it does not communicate address and data bits of PCI bus transaction in serial form over that channel. Ex. 2021 at ¶¶ 100-02, 132-38. Horst's use of the TNet protocol means that PCI address bits are not transmitted over the TNet links. Horst defines the packet format in figure 5. Ex. 1009 at Figure 5. A PCI address is not included. *Id.*

At best, Horst transmits information that specialized hardware and software on the peripheral side can use later to generate a PCI bus transaction. Ex. 2021 at

¶¶ 138, 141. That PCI bus transaction is then communicated only in *parallel* form. Accordingly, Horst fails to teach this limitation. And no other reference is put forward to teach this portion of the claimed invention.

Consequently, neither Horst, nor any combination with Horst, teaches the system claimed in claim 24. Horst teaches a different architecture and never serializes an actual PCI bus transaction. Ex. 2021 at ¶¶ 100-02, 132-38.

Finally, as Dr. Lindenstruth confirms, it would not be obvious to communicate using PCI over the TNet link. Ex. 2021 at ¶¶ 143-52. First, Horst teaches away from using PCI transactions over the TNet link. Ex. 1009 at 1-2. Horst explicitly considered standard busses like PCI and discarded them in favor of a completely new proprietary network. *Id.* Second, communicating PCI over TNet would render the system inoperable because the memory structure of TNet is incompatible with PCI transactions. Ex. 2021 at ¶¶ 148-50. Third, even if the system was not rendered inoperable, communicating PCI bus transactions over TNet would add unnecessary complexity and expense to a functioning system. *Id.* at ¶ 151.

The subsections below address these points in more detail.

a. Horst does not disclose a "north bridge to communicate address and data bits of PCI bus transaction in serial form"

Claim 24 recites, in part:

> "a north bridge to communicate address and data bits of PCI bus transaction in serial form"

Ex. 1001, claim 24. This claim limitation specifically requires both address and data bits of PCI bus transaction be communicated in serial form. However, nothing in Horst discloses a north bridge that communicates address bits of *PCI* bus transaction in serial form. Ex. 1009 at 4-7, Figs. 5, 7; Ex. 2021 at ¶¶ 100-02, 132-38.

(1) Horst's "north bridge" is directly coupled to the microprocessor

Petitioner asserts that Horst discloses a north bridge in the form of a TNet processor interface, which is directly coupled to the microprocessor. Paper No. 2 at 31. The excerpted and annotated figures at Figure E below show the relationship of the TNet processor interface in relation to the overall system architecture. Horst's figure 9 (on the left) shows the high level architecture of a typical system using TNet, including the processor interface. Ex. 1009 at 8, Fig. 9. Horst's figure 7 shows a detailed view of that processor interface. Ex. 1009 at 6, Fig. 7.

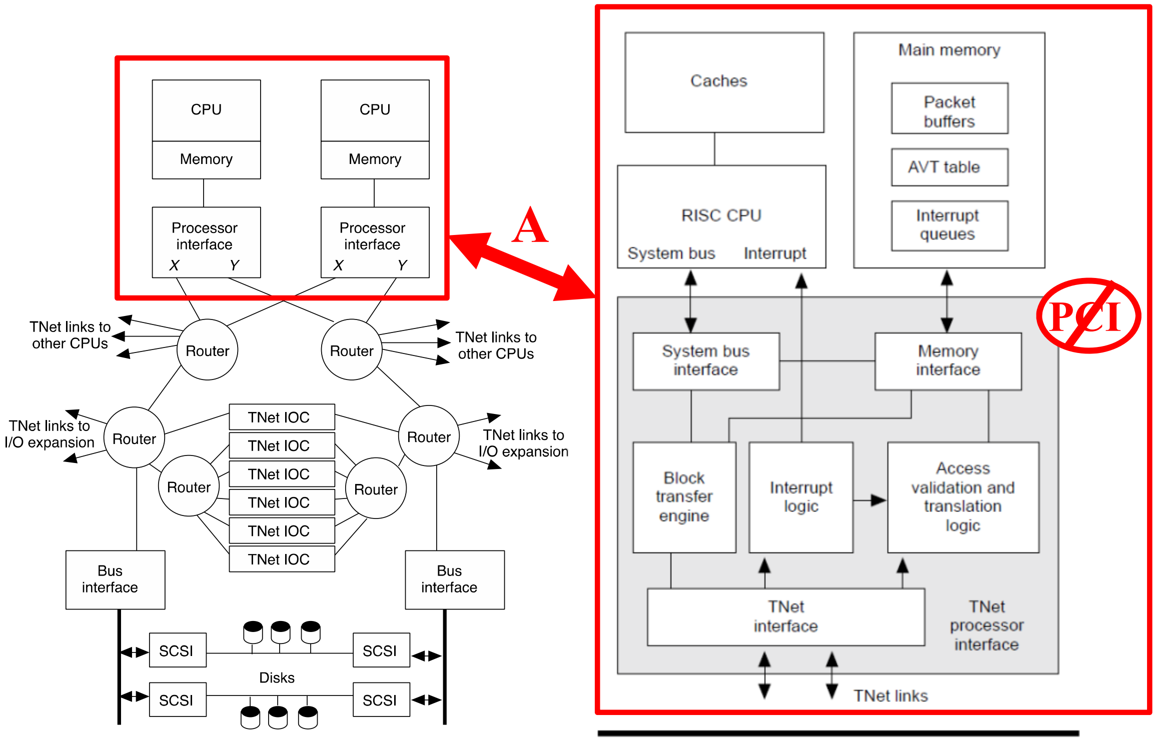


Figure 9. Typical system architecture using TNet.

Figure 7. TNet processor interface.

Figure E: TNet Processor Interface in Relation to the Typical TNet Architecture

As shown by the arrow labelled "A," the TNet processor interface is directly coupled to the CPU-memory bus. There are no other CPU-side interfaces or busses described in Horst.

(2) Horst's "north bridge" does not communicate address bits of PCI bus transaction

CPU Side

The TNet processor interface does not communicate address bits of *PCI* bus transaction as required by claim 24. Ex. 2021 at ¶¶ 100-02, 132-42, 145, 148-51. To find that the north bridge communicates address bits of PCI bus transaction in

Horst, a PCI transaction must be found in the TNet processor interface. *Id.* at ¶¶ 129-30. As shown in Figure F, the TNet processor interface does not communicate address bits of PCI bus transactions. Ex. 1009 at 5-6, Figs. 5, 7.

The PCI standard sets out specific requirements for PCI standard addresses. Ex. 2001 at 25, 37-39, 42; Ex. 2021 at ¶¶ 61-72. Critically, all PCI addresses are physical addresses in one of three memory spaces. Ex. 2001 at 42; Ex. 2021 at ¶¶ 61-72, 137. The only addresses disclosed in Horst that are communicated over TNet are virtual TNet addresses. Ex. 1009 at 5, Fig. 5; Ex. 2021 at ¶¶ 95, 101-04. The format of a TNet packet is shown in figure 5 of Horst, excerpted and annotated below as Figure F.

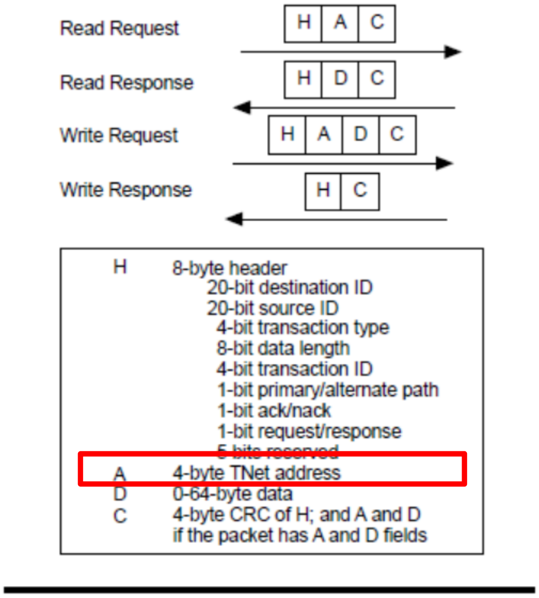


Figure 5. TNet read and write transactions.

Figure F: The TNet Packet

Horst states that TNet addresses are virtual addresses, not physical addresses. Ex. 1009 at 2, 5-6; Ex. 2021 at ¶¶ 95, 101-04. This means that TNet addresses cannot be PCI standard addresses because physical addresses cannot be virtual addresses. Ex. 2021 at ¶¶ 38, 39, 134-37. Upon configuration, a PCI bus master must map the PCI address of each PCI device into a specific physical address location in the processor's memory or I/O address space. *Id.* at ¶ 64. This is a flat addressing scheme, which is an addressing scheme for a single processor node address space. Ex. 2015 at 9, 12; Ex. 2016 at 6; Ex. 2021 at ¶¶ 60-64. TNet addresses do not use a flat address space because TNet interconnects multiple processor nodes, each with its own address space. Ex. 1009 at 7. Therefore, the flat addressing scheme used by PCI would not work on TNet. Ex. 2016 at 6; Ex. 2017 at 3-4; Ex. 2020 at 3; Ex. 2021 at ¶¶ 135-37, 148-51.

Also, because the TNet processor interface adapter intakes only CPU-memory bus transactions on the system bus interface, there is never a PCI bus transaction on the CPU side in the first place. *Id.* at ¶ 95. Instead, the TNet processor interface translates a CPU-memory bus transaction into a TNet packet for transmission on the TNet link. Figure F; Ex. 2021 at ¶¶ 95, 100, 129. This is highlighted by the "Access validation and translation logic" block that handles incoming and outgoing traffic. Ex. 2021 at ¶ 141.

Peripheral Side

Switching now to the peripheral side, the presence of the TNet bus interface confirms that the TNet processor interface does not carry a PCI bus transaction. The excerpted and annotated figures at Figure H show the relationship of the TNet bus interface in relation to the overall system architecture with the arrow labeled "B." Horst's figure 9 (on the left) shows the high-level architecture of a typical system using TNet. Horst's figure 8 (on the right) shows a detailed view of the peripheral side of a TNet system.

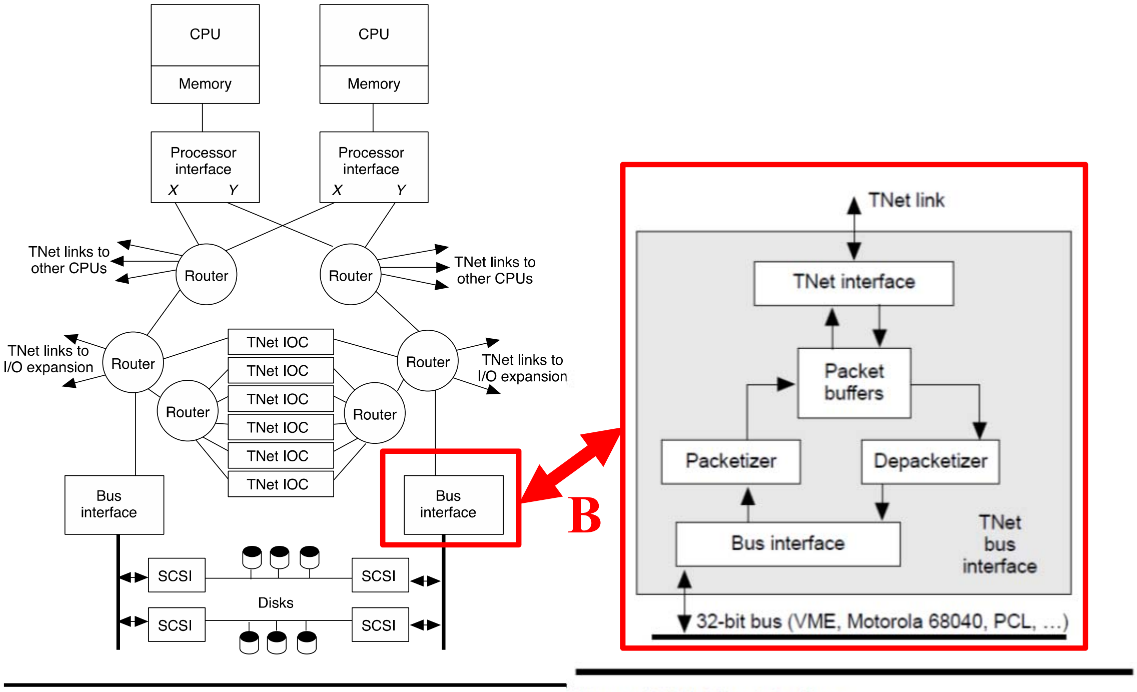


Figure 9. Typical system architecture using TNet.

Figure 8. TNet bus interface.

Figure G: TNet Bus Interface in Relation to the Typical TNet Architecture

The TNet bus interface (right) receives the TNet packet over the TNet link and translates it into a bus transaction for whichever 32-bit bus it is coupled to (e.g., VME, PCI, or SCSI). The TNet bus interface uses an intelligent ASIC, which a POSA would understand generates the appropriate bus commands and addresses based on the particular bus configuration. Ex. 2021 at ¶ 138. As a result, the TNet transaction need not be a serialized PCI standard bus transaction for the intelligent TNet PCI interface ASIC to generate a PCI bus transaction on the parallel PCI bus. *Id.* at ¶ 133. That the TNet bus interface is intelligent makes sense because TNet needs to interact with multiple processors and bus interfaces, each connecting to a different bus protocol with different devices. *Id.*; also Ex. 2016 at 6; Ex. 2017 at 3-4; Ex. 2020 at 3.

The system works similarly in reverse. In that case, the TNet bus interface receives a PCI transaction from the peripheral-side PCI bus which is parallel, not serial. Ex. 2021 at ¶ 94. The intelligent ASIC in the TNet bus interface generates a TNet transaction for transmission over the TNet link. *Id.* at ¶¶ 94, 133-138. For all the reasons discussed herein, the TNet bus interface does not transmit PCI address and data bits in serial form over the TNet link. Supra at § IV(A)(1). It simply would not work.

Note that because Horst removed the CPU-side PCI bus, extensive specialized hardware and software (the TNet bus interfaces) are required for TNet to function. Ex. 2021 at ¶¶ 96-98.. Therefore, modified drivers are required for all peripheral devices. *Id.* As discussed above, a key advantage to the claims of the '814 is the compatibility with legacy devices. *Id.* at 97; *supra* at II(C) TNet eliminates this advantage.

The presence of the TNet bus interface on the peripheral side of Horst does not demonstrate that the TNet link carries address and data bits of PCI bus transaction in serial form, as recited in claim 24. Ex. 2021 at ¶ 133. In fact, it shows the opposite. The TNet bus interface would not be necessary if the TNet link were capable of carrying PCI address bits. Horst's TNet bus interface is required to convert the TNet information coming off of the TNet link from the CPU into the PCI protocol format, including by adding/appending the PCI address so that the data can be carried on the only PCI bus in the TNet system: the peripheral-side, parallel bus. Ex. 2021 at ¶ 141. For the information coming off of Horst's peripheral-side, parallel PCI bus into the TNet bus interface, the TNet bus interface is necessary to strip away the PCI information, such as the PCI address, and convert the data to the format that is supported by Horst's TNet link, which does not include the required information to qualify as a PCI bus transaction (e.g., a PCI address). Ex. 2021 at ¶¶ 101-02. Horst's TNet bus interface even requires special hardware—an ASIC—to accomplish this conversion because Horst's TNet link—the only serial bus in Horst—cannot carry address and data bits of PCI bus transaction in serial form. Ex. 2021 at ¶¶ 138.

TNet does not use PCI for its entirely new and proprietary network. The new network (TNet) described by Horst uses virtual addressing and address translation and purposefully avoids memory-mapped I/O and physical addressing (e.g., PCI) required for a PCI bus transaction. Ex. 1009 at 2, 5-6; Ex. 2021 at ¶¶ 95, 101-04. The TNet packets and transaction types do not allow for the PCI standard bus transactions, including memory read and write, I/O read and write, and configuration read and write. Ex. 1009 at 4-5, Fig. 5, Ex. 2001 at 21; Ex. 2021 at ¶ 104. Nor does Horst disclose any mechanism for the TNet packets to communicate PCI bus commands indicating a type of PCI standard bus transaction. Ex. 1009 at 4-5; Ex. 2001 at 21; Ex. 2021 at ¶ 104. All these differences make clear that TNet simply does not match the claim limitations that require a "north bridge to communicate address and data bits of PCI bus transaction in serial form."

<u>Horst Moved the PCI Interface to the Peripheral Side</u>

Figure H (excerpted and annotated figures 1 & 2 from Horst), below, illustrates that Horst designed the TNet system to specifically remove the PCI bus from the CPU side to be able to communicate between CPUs. Horst Figure 1 (left) shows a traditional, PCI-based system. Ex. 1009 at 2, Fig. 1. Horst Figure 2 (right) shows a typical TNet architecture. Ex. 1009 at 2, Fig. 2. Horst had to replace the PCI bus in the traditional system because it is incapable of performing the CPU-to-CPU communication Horst desired. Ex. 2016 at 6; Ex. 2017 at 3-4;

Ex. 2020 at 3; Ex. 2021 at ¶¶ 1133-142, 145, 148-151. Horst replaced the CPU-side PCI bus with the TNet link. *Compare* Ex. 1009 Fig. 1 *with* Ex. 1009 Fig. 2. Where, in the traditional system (left), the PCI interface existed on the CPU side, Horst intentionally moved the PCI interface into the peripheral side in the TNet system (right), where it resides at the same level as all the other bus interfaces.

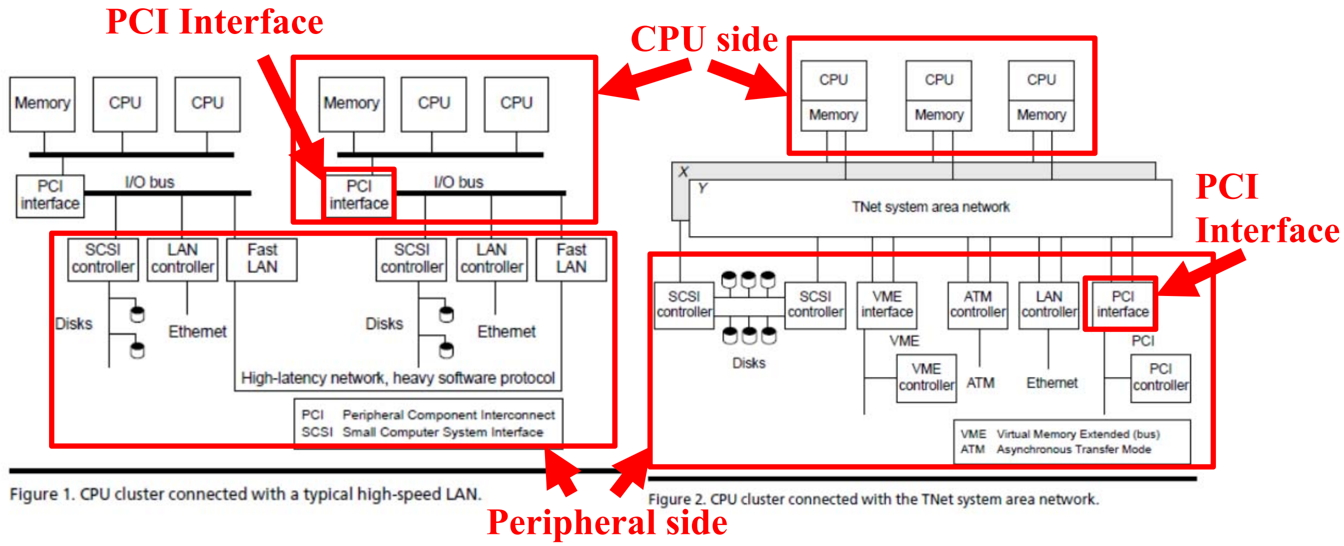


Figure 1. CPU cluster connected with a typical high-speed LAN.

Figure 2. CPU cluster connected with the TNet system area network.

Figure H: Comparison of Traditional PCI-based Architecture and TNet Architecture

The TNet processor interface (shown in detail in Figure E above) is directly coupled to the CPU-memory bus and therefore does not start with a PCI bus transaction to translate. Additionally, the TNet processor interface translates the CPU-memory bus transaction into a TNet packet, which does not include PCI address bits. Ex. 1009 at 4-5, 7, Fig. 5; Ex. 2021 at ¶¶ 133-142, 145, 148-151. Instead, the TNet bus interface ASIC generates the parallel PCI bus transaction on the receiving end for transmission on the parallel PCI bus. Ex. 2021 at ¶¶ 133-142, 145, 148-151. Therefore, the TNet processor interface, which is on the CPU side, does not communicate address bits of PCI bus transaction.

<u>Petitioner's Arguments Do Not Account for the Claim Requirements</u>

Petitioner and Mr. Young appear to argue that, since a PCI bus transaction occurs at some point in the system on the peripheral side, every part of that system is communicating or transmitting that PCI bus transaction. Ex. 1003 at 98, 104-107. The presence of a PCI bus interface on the peripheral side does not mean that the TNet processor interface on the CPU side arrives as a PCI bus transaction. Ex. 2021 at ¶ 133. In fact, the TNet processor interface cannot carry a PCI bus transaction as recited in claim 24. *Supra* at § IV(A)(1)(a)(2). The presence of the PCI bus interface on the peripheral side means only that the peripheral side can handle parallel PCI bus transactions. Ex. 2021 at ¶¶ 133-142, 145, 148-151.

Further, as a practical matter, Mr. Young's conclusion is illogical. By the same logic employed by Mr. Young, the presence of other peripheral interfaces, like the VME interface, would mean that the TNet processor interface is also carrying VME bus transactions—and all other types of transactions associated with any communication protocol shown as attached to the TNet network in Horst.

As already discussed, the PCI address bits are not communicated over the TNet link, and the PCI address is critical to a PCI bus transaction. Ex. 2021 at ¶¶ 133-142, 145, 148-151. The reason that it is not necessary to transmit the PCI address over the TNet link is because of the TNet bus interface. Dr. Lindenstruth explains that the TNet bus interface, as an intelligent ASIC, performs the necessary translation and processing to generate the appropriate transaction for the bus or network it is transmitting the instruction over. *Id*. at ¶¶ 133-138.

(3) Horst's "north bridge" does not communicate address bits of PCI bus transaction in serial form

The TNet processor interface (Horsts "north bridge") does not communicate address bits of PCI bus transaction in serial form. As noted above, the TNet processor interface does not communicate address bits of PCI bus transaction at all. *Supra* at § IV(A)(1)(a)(2).

Notably, the PCI bus transaction is not generated until the TNet packet is received by the TNet bus interface for a PCI bus on the peripheral side. Ex. 2021 at ¶¶ 133-142, 145, 148-151. Once received, the TNet bus interface ASIC generates the PCI bus transaction, (*see id*. at ¶ 133-138) which is the first time that the PCI bus transaction exists in the system. *Id*. at ¶¶ 133-142, 145, 148-151. At that time, the PCI bus transaction is communicated *in parallel form* over the PCI bus. *Id*. at ¶ 138; Ex. 1009 at 2 Fig. 2.

Dr. Lindenstruth confirms that Horst does not disclose a "north bridge to communicate address and data bits of PCI bus transaction in serial form." Ex. 2021 at ¶¶ 133-142, 145, 148-151. Accordingly, Horst does not disclose or render obvious each limitation of claim 24. Therefore, this Board should confirm claims 24 over Ground 1.

b. Horst does not disclose a "second LVDS channel extending from said north bridge to convey said address and data bits of PCI bus transaction"[3]

Turning to another independent reason that claim 24 is not anticipated or rendered obvious by Ground 1, claim 24 recites, in part:

> "said second LVDS channel extending from said north bridge to convey said address and data bits of PCI bus transaction in serial form"

As explained above, BRI of "extending from" is "directly coupled to." Therefore, the LVDS channel identified, and the only possible LVDS channel that Petitioner could identify, is the TNet link. As shown in Figure E (above), only the TNet link is directly coupled to the TNet processor interface on the CPU side. Ex. 1009 at 2, 6, 8, Figs. 2, 7, 9; *also supra* at § IV(A)(1)(a)(1) Fig. E. However, as explained in detail in Section IV(A)(1)(a) above, the TNet processor interface does not communicate a PCI bus transaction or the address a bits of a PCI bus

[3] ACQIS does not dispute the presence of a low voltage differential signal (LVDS) channel, but does dispute the Board's construction and does not acquiesce to it. The construction of a low voltage differential signal channel has not been argued and its presence is not relevant to ACQIS's validity arguments.

transaction in serial form over the TNet link. Therefore, Horst does not disclose this limitation.

Instead, the only PCI address bits generated anywhere in the TNet system are when the remote parallel PCI bus transaction is generated by the TNet bus interface and transmitted over the PCI bus *on the peripheral side*. *See supra* at § IV(A)(1)(a)(3). The PCI bus is parallel, so the PCI address bits are never conveyed in serial form in the TNet system. *Supra* at § IV(A)(1)(a)(3); Ex. 2021 at ¶ 138.

Accordingly, as Dr. Lindenstruth confirms, Horst does not disclose a "north bridge to communicate address and data bits of PCI bus transaction in serial form." Ex. 2021 at ¶¶ 130-138. Further, neither Horst alone nor in combination with any secondary reference disclose or render obvious this limitation, and Petitioner has not argued that they do. Ex. 2021 at ¶¶ 143-152. For that reason, this Board should find that Ground 1 does not invalidate claim 24.

c. Conclusion: Claim 24 is Valid over Horst

As detailed above in Section IV.A.1, Horst does not disclose a "north bridge to communicate address and data bits of PCI bus transaction in serial form." Horst shows, and Dr. Lindenstruth confirms, that the TNet bus interface does not communicate PCI address bits. Ex. 2021 at ¶¶ 130-142. Further, Horst does not disclose a "second LVDS channel extending from said north bridge to convey said address and data bits of PCI bus transaction." *Id.* Horst does not disclose that TNet can, and Dr. Lindenstruth confirms that TNet cannot, convey address bits of PCI bus transaction. *Id.* at ¶¶ 130-142, 145, 148-51. Accordingly, claim 24 is valid over Horst and this Board should confirm claim 24 over Ground 1.

2. Claim 31 is not rendered obvious over Horst, Mathers, and the LVDS Owner's Manual (Ground 1)

Like claim 24, claim 31 requires a "second LVDS channel to communicate address and data bits of PCI bus transaction." Ex. 1001, claim 31. Again, Horst does not teach this limitation. As set forth in more detail below, Horst does use a serial communication link between the computer and the peripheral devices, but it does not communicate address and data bits of PCI bus transaction in serial form over that link. Ex. 2021 at ¶¶ 130-42; *supra* at § IV(A)(1)(a). At best, Horst transmits information that specialized hardware and software on the peripheral side can use later to generate a PCI bus transaction. Ex. 2021 at ¶¶ 138; *supra* at § IV(A)(1)(a)(3). That PCI bus transaction is then communicated only in parallel form to PCI devices. Ex. 2021 at ¶¶ 130-42. And no other reference is put forward to teach this portion of the claimed architecture.

Accordingly, neither Horst nor any combination with Horst teaches the system claimed in claim 31. Horst teaches a different architecture and never serializes an actual PCI bus transaction as required for claim 31. At best, Horst generates a parallel PCI bus transaction on the remote peripheral side. *Supra* at §§ IV(A)(1)(a)(3).

a. Petitioner points to TNet Link as the second LVDS channel

Petitioner points to the TNet link, which extends from the TNet processor interface, as the second LVDS channel for the "second LVDS channel" limitation of claim 31. Paper No. 2 at 33. The TNet link relied on by Petitioner connects the CPU side with other devices on the TNet link, including other computers, I/O, and peripheral devices.

b. Horst does not disclose a "second LVDS channel to communicate said address and data bits of PCI bus transaction"

The TNet link is the only LVDS channel disclosed in Horst. The TNet link, however, does not communicate address and data bits of PCI bus transaction. *Supra* at §IV(A)(1)(a)(2)-(3).

First, the only addresses disclosed in Horst that are communicated over TNet are virtual TNet addresses. *Supra* at §IV(A)(1)(a)(2)-(3); Ex. 1009 at 5, Fig. 5; Ex. 2021 at ¶¶ 130-42. TNet addresses are virtual rather than physical addresses, as the PCI standard requires. Ex. 1009 at 2, 5-6; Ex. 2021 at ¶ 100. Second, no PCI bus transaction is transmitted over the TNet link (and, consequently, the PCI address bits are not communicated over the TNet link) to the remote, peripheral-side TNet bus interface. Ex. 2021 at ¶¶ 133-38. The TNet-to-PCI bus interface is an intelligent ASIC that generates PCI bus transactions using the received TNet packets and the information about the PCI bus contained in the ASIC. *Id.* at ¶¶ 139, 142. The generated PCI bus transaction is only carried by a parallel bus on the peripheral side. Ex. 2021 at ¶ 138. Third, TNet cannot transmit address bits of PCI bus transaction because it does not support memory-mapped I/O, it uses virtual addresses, and it does not use the PCI bus command information required for a PCI bus transaction. These distinctions are discussed in greater detail above in Section IV(A)(1)(a).

For the foregoing reasons, Horst does not disclose an LVDS channel to communicate address and data bits of PCI bus transactions. Ex. 2021 at ¶¶ 130-42, 145, 148-151. Therefore, Petitioner has failed to meet their burden to show that claim 31 is invalid. Additionally, Dr. Lindenstruth confirms that it would not be obvious to modify Horst to communicate address and data bits of PCI bus transaction and Petitioner does not argue that it would. Ex. 2021 at ¶¶ 143-151. Finally, neither Mathers nor the LVDS manual teach or suggest this limitation and Petitioner does not suggest that they do. Accordingly, this Board should confirm claim 31 over Ground 1.

3. Claims 32 and 33 are also valid over Horst (Ground 1)

Claims 32 and 33 each depend from claim 31. Because the references in Ground 1 do not render obvious or disclose every limitation of claim 31, claims 32 and 33 are also not disclosed or rendered obvious.

4. Claims 31-33 are not rendered obvious over Bogaerts, Gulick, Mathers, and James (Ground 2)

Claim 31 requires a "second LVDS channel to communicate address and data bits of PCI bus transaction in serial form." Ex. 1001 at cl. 31. Dr. Lindenstruth, an author of the Bogaerts reference confirms it does not teach this limitation. Ex. 2021 at ¶¶ 153-161. Bogaerts is Petitioner's only institute reference for this limitation. Paper No. 11 at 20.

Bogaerts uses a serial communication protocol (SCI) to interconnect computer processor nodes and remote peripheral devices, but it does not communicate CPU-side address and data bits of PCI bus transactions in serial form. Ex. 1011 at 15-17, Fig. 13; Ex. 2021 at ¶¶ 107-110, 153-161. Bogaerts eliminated the CPU-side PCI bus in favor of an SCI ring network to allow massively-parallel computing. Ex. 2021 at ¶¶ 107-110. At best, Bogaerts transmits information that specialized hardware and software on the remote peripheral side can use to generate a PCI bus transaction. Ex. 1011 at 16, Figs. 13-14; Ex. 2021 at ¶¶ 107-110, 153-161. That PCI bus transaction is then communicated only in parallel form over a traditional PCI bus. *Id.* And no other reference was instituted to teach this portion of the claimed architecture.

Accordingly, Dr. Lindenstruth confirms, neither Bogaerts nor any combination with Bogaerts teaches the system claimed in claim 31. Ex. 2021 at ¶¶ 153-69. Bogaerts teaches a different architecture and never serializes an actual PCI bus transaction. *Id.* At best, Bogaerts generates a parallel PCI bus transaction on the remote peripheral side, which is transmitted in parallel form. Id. at ¶¶ 107-111, 153-69.

The subsections below address these points in more detail.

a. Petitioner points to the SCI ring as the second LVDS channel

Petitioner points to the SCI ring of Bogaerts as the second LVDS channel for the "second LVDS channel" limitation. Paper No. 2 at 47. And, the SCI ring is the only LVDS serial channel disclosed by Bogaerts. *See* Ex. 1011 at 16, Fig. 13.

b. Bogaerts does not disclose a "second LVDS channel to communicate said address and data bits of PCI bus transaction"

As Dr. Lindenstruth confirms, the SCI ring disclosed in Bogaerts does not communicate "address and data bits of PCI bus transaction." Ex. 2021 at ¶ 156. The PRO-SCI adapter and the PCI-SCI adapter are disclosed as nodes on the SCI ring. Ex. 1011 at Figs. 13 and 14. The PRO-SCI adapter communicates CPU-side SCI transactions. Ex. 2021 at ¶ 160. The PCI-SCI adapter communicates peripheral-side SCI transactions. *Id.* at ¶¶ 155-56. And, as Dr. Volker explains, neither of the adapters communicates address bits of PCI bus transaction over the SCI ring. *Id.* at ¶¶ 155-56, 160-61.

The PCI-SCI adapter disclosed in Bogaerts is the bridge that is necessary to couple the peripheral-side SCI ring to a remote PCI parallel bus. *Id.* at ¶ 159. A PCI-SCI adapter is necessary because the SCI transactions on the SCI ring are not PCI transactions and are not compatible with a PCI bus. *Id.* at ¶ 155-56. Specifically, a PCI address is much different than an SCI address. SCI uses a 64-bit address space as opposed to the 32-bit address space primarily used by the PCI standard. *Id.* at ¶¶ 108-09. The protocol differences require the PCI-SCI adapter to translate the address and transaction information as described below.

Dr. Lindenstruth, an author on the Bogaerts reference and one of the chief designers of the CERN/LBL PCI-SCI adapter discussed in Bogaerts, confirms that the PCI-SCI adapters disclosed in Bogaerts do not transmit PCI address bits over the SCI ring. *Id.* at ¶ 160. Instead, even when a PCI transaction is received by the PCI-SCI adapter, it discards the PCI bus commands and a portion of the PCI address, all of which is required for a PCI bus transaction. *Id.* at ¶ 109, 155. The intelligent PCI-SCI adapter then translates the remainder of the PCI address into a 64-bit SCI address and selects the appropriate SCI transaction type for transmission over the SCI ring. *Id.* Because an SCI address is not a PCI address, the PCI-SCI adapter does not transmit address bits of PCI bus transaction over the CPU-side SCI ring. *Id.* at ¶ 156.

The other type of SCI adapter disclosed by Bogaerts is the PRO-SCI adapter. As shown in Figure 8, the PRO-SCI adapter couples the CPU-memory bus to the SCI ring. *Id.* at ¶ 160. Because the PRO-SCI adapter takes in only CPU-memory bus transactions, there is never a PCI bus transaction on the CPU-side PRO-SCI adapter. *Id.* at ¶ 156. Dr. Lindenstruth confirms that the PRO-SCI adapter generates an SCI transaction to be sent over the CPU-side SCI ring. *Id.* at ¶ 160.

Nothing in Bogaerts suggests that address bits of PCI bus transaction are transmitted on the SCI ring by a PRO-SCI adapter. In fact, PCI address bits are not transmitted by the PRO-SCI adapter, which allows memory-to-memory access between multiple processor nodes and connection to remote I/O devices. *Id.* at ¶¶ 111, 159-60. The flat PCI address space cannot be used to interconnect multiple processor nodes like the PRO-SCI adapter of Bogaerts allows because it will result in address collisions and conflicts. *Id.* at ¶ 76; also Ex. 2016 at 6; Ex. 2017 at 3-4; Ex. 2020 at 3. Further, as disclosed in Bogaerts, the PRO-SCI adapter does not include the PCI-SCI ASIC, which is necessary to generate a PCI transaction from an SCI transaction. Ex. 1011, Figs. 8, 14.

On the remote, peripheral side, the PCI-SCI adapter translates the SCI transaction into a remote PCI bus transaction which is transmitted in parallel form over the PCI bus. Ex. 1011 at 16 Fig. 13, 17 Fig. 15; Ex. 2021 at ¶¶ 153-61. The PCI bus transaction is never transmitted over a serial channel.

Note that because Bogaerts removed the CPU side PCI bus, extensive specialized hardware and software (the PCI-SCI adapter) are required for Bogaerts to function. Ex. 1011 at 16 Figs. 13, 14; Ex. 2001 at 19, Fig. 1-2; Ex. 2021 at ¶ 155. Therefore, modified drivers are required for all peripheral devices. *Id.*; Ex. 2014 at 3. As discussed above, a key advantage to the claims of the '814 is that the compatibility with legacy devices. Ex. 2012 at 20; Ex. 2014 at 3; Ex. 2018 at 4:50-58; Ex. 2021 at 97, 125; supra at § II(C)(2). Bogaerts eliminates this advantage and instead opts for parallel processing ability.

The presence of the PCI-SCI adapter on the peripheral side of Bogaerts does not demonstrate that the SCI ring carries address and data bits of PCI bus transaction in serial form as recited in claim 31. Ex. 2021 at ¶¶ 155-56. In fact, it shows the opposite. The PCI-SCI adapter would not be necessary if the SCI ring were capable of carrying PCI address bits. Bogaerts' PCI-SCI adapter is required to convert the SCI information coming off of the SCI ring from the CPU into the PCI protocol format, including by adding/appending the PCI address so that the data can be carried on the only PCI bus in the Bogaerts' system: the peripheral-side, parallel bus. *Id.* For the information coming off of Bogaerts' peripheral-side, parallel PCI bus into the PCI-SCI adapter, the PCI-SCI adapter is necessary to strip away the PCI information, such as the PCI address, and convert the data to the format that is supported by Bogaert's SCI ring, which does not include the required information to qualify as a PCI bus transaction (e.g., a PCI address). *Id.* at ¶ 155. Bogaert's PCI-SCI adapter even requires special hardware—an ASIC—to accomplish this conversion because Bogaert's SCI ring—the only serial bus in Bogaerts —cannot carry address and data bits of PCI bus transaction in serial form. Ex. 1011 at Figs. 13, 14.

Consequently, as Dr. Lindenstruth confirms, Bogaerts does not disclose an LVDS channel that communicates address and data bits of PCI bus transaction in serial form.[4] Ex. 2021 at ¶ 156.

c. Neither James nor Gulick remedy the problem with Bogaerts

The combination of Bogaerts with James and Gulick does not teach a "second LVDS channel to communicate said address and data bits of PCI bus transaction." Neither James nor Gulick include a suitable LVDS channel that meets the requirements of the "second LVDS channel."

[4] ACQIS does not dispute the presence of a low voltage differential signal (LVDS) channel, but does dispute the Board's construction and does not acquiesce to it. The construction of a low voltage differential signal channel has not been argued and its presence is not relevant to ACQIS's validity arguments.

(1) Petitioner provides no rationale to combine Bogaerts with either James or Gulick

As an initial matter, as argued in the Preliminary Response, Petitioner has not provided a sufficient rationale for combining either James or Gulick with Bogaerts. ACQIS' arguments against combining the references is set forth in its Preliminary Response. Paper No. 7 at 42-45. The Board did not rely on either James or Gulick in the Decision. Paper No. 11 at 20. But, in an abundance of caution, they are addressed here anyway.

(2) James fails to teach a second LVDS channel communication serial address bits of PCI transaction for the same reasons as Bogaerts.

The James reference discloses a similar system to Bogaerts, with a multiple processor node memory-to-memory using a newer version of SCI. Ex. 2021 at ¶¶ 162-166. As a result, James does not disclose transmitting serial address bit of a PCI transaction for the same reason Bogaerts does not—the system would not function. *Id*.

<u>**James' LVDS Channel Does Not Support PCI Transactions**</u>

The primary difference between James and Bogaerts is that James uses SerialExpress to provide memory-to-memory access between processor nodes instead of SCI. Ex. 2021 at ¶ 165; Ex. 1018 at FIGs. 1, 6; 6:23-26; Ex. 2016 at 6; Ex. 2017 at 3; Ex. 2020 at 3. SerialExpress is the next iteration of the SCI standard. Ex. 2021 at ¶ 165. And like the SCI ring in Bogaerts, the SerialExpress network in James cannot transmit address bits of PCI bus transaction in serial form for the same reason—the SerialExpress address space does not support the PCI address space or PCI transaction types. Ex. 2021 at ¶¶ 162-66. Further, because the James SerialExpress link is a memory-to-memory system, PCI transactions would not work due to address collisions. *Id.*; Ex. 2016 at 6; Ex. 2017 at 3; Ex. 2020 at 3; Ex. 2015 at 12.

Mr. Young's claim that the transaction transmitted by the James host adapter to the remote PCI bus would "necessarily" include PCI bus transaction address and data bits is wrong. Ex. 1003 at 131-32; Ex. 2021 at ¶ 165-66. Mr. Young cites no support for this conclusory statement—rendering Petitioner's arguments regarding James improper, unsupported inherency arguments.

<u>No CPU-Side PCI Transaction</u>

Similar to the PRO-SCI adapter of Bogaerts, the James' host adapter receives a CPU-memory bus transaction and transmits a SerialExpress transaction over the SerialExpress channel. Ex. 1018 at FIGs. 1, 6, 6:23-26. Further, like the PRO-SCI adapter described in Bogaerts, nothing in James suggests that the host adapter communicates address and data bits of PCI bus transaction. *Id.*

<u>Peripheral-Side PCI Transaction is Parallel</u>

Again, similar to the Bogaerts system, the only PCI transaction disclosed in James is after the SerialExpress link goes through a SerialExpress-to-PCI adapter on the peripheral side, (James at FIG. 6, element adaptA 42a.) Ex. 1018 at Fig. 6. The PCI transaction generated on the peripheral PCI bus by the SerialExpress to PCI adapter and is a parallel transaction on the remote PCI bus. Ex. 2001 at 17; Ex. 2021 at ¶ 165-66.

(3) Gulick's System Does Not Teach a Second LVDS that communicates serial address bits of a PCI bus transaction.

Like Bogaerts and James, Gulick removes the PCI bus between north bridge and south bridge and places it in the south bridge. Ex. 1010 at Fig. 2; Ex. 2021 at ¶¶ 167-69. Gulick uses a high velocity-low pin count ("HVLPC") bus to replace the PCI bus found between the north bridge and the south bridge in traditional systems (see Figure A, above). Ex. 1010 at Figs. 2, 3:47-67; Ex. 2021 at ¶ 167. However, as seen in figure 2 of Gulick, that system designer simply moved the original PCI bridge that normally resides on the north bridge to the south bridge. Ex. 1010 at Fig. 2 (element 127).

<u>No CPU-side PCI Transaction</u>

Like TNet and Bogaerts, there is no disclosure of any PCI bus transactions on the CPU side. Ex. 2021 at ¶¶ 167-69. Instead, Gulick's north bridge takes in a processor memory bus transction and converts it directly to a HV-LPC transaction. Ex. 1010 at Fig. 2; Ex. 2021 at ¶¶ 167-69. There is no indication or reason that a PCI transaction ever occurs CPU-Side in Gulick. *Id.*

Peripheral Side PCI Transaction is Parallel

Also like TNet and Bogaerts, the only PCI bus transaction disclosed in Gulick is a parallel PCI bus transaction on the peripheral side of the HV-PLC I/F controller in the south bridge. Ex. 1010 at Fig. 2 (element 127); Ex. 2021 at ¶¶ 167-69. As stated previous, simply because a PCI transaction is generated on the peripheral side from an intelligent ASIC does not mean there was an intermediate PCI transaction. Ex. 2021 at ¶¶ 133, 166; *see supra* at §IV(A)(1).

Indeed, the presence of the HV-LPC I/F controller in the south bridge indicates the HV-LPC bus is not a PCI transaction because it connects to multiple separate busses. *Id.* at ¶¶ 167-69. Therefore, as Dr. Lindenstruth confirms, James does not disclose an LVDS channel that communicates address and data bits of PCI bus transaction in serial form.

(4) The Gulick Local Interconnect System is Incompatible with Bogaerts SCI Ring

Additionally, a POSA would not combine Gulick with Bogaerts. Ex. 2021 at ¶ 169. The HV-LPC bus described in Gulick is local interconnection between the North Bridge and South Bridge in a *single* computer system. Ex. 1010 at Fig. 2. Bogaerts, on the other hand, replaces the connection between the traditional north bridge and south bridge with an SCI ring capable of remote connections and memory-to memory transactions between processor nodes. Ex. 1011 at 16 Fig. 13.

There is no evidence to suggest the Gulick system's local north bridge to south bridge HV-LPC connection could support Bogaerts' purpose of interconnecting *multiple* processor nodes and remote I/O devices. Ex. 2021 at ¶ 167. As a result, a POSA would not combine Gulick with Bogaerts because they disclose incompatible systems.

Because neither Gulick or James disclose a peripheral bridge that communicates address and data bits of PCI bus transaction, Petitioner has failed to show that this limitation is rendered obvious over the proposed combination of Bogaerts, James, and Gulick. Further, Petitioner has failed to provide sufficient rationale for combining the three references. Therefore, the Board should confirm claim 31 over Ground 2.

d. Claims 32 and 33

Claims 32 and 33 each depend from claim 31. Because the references in Ground 2 do not render obvious or disclose every limitation of claim 31, claims 32 and 33 are also not disclosed or rendered obvious.

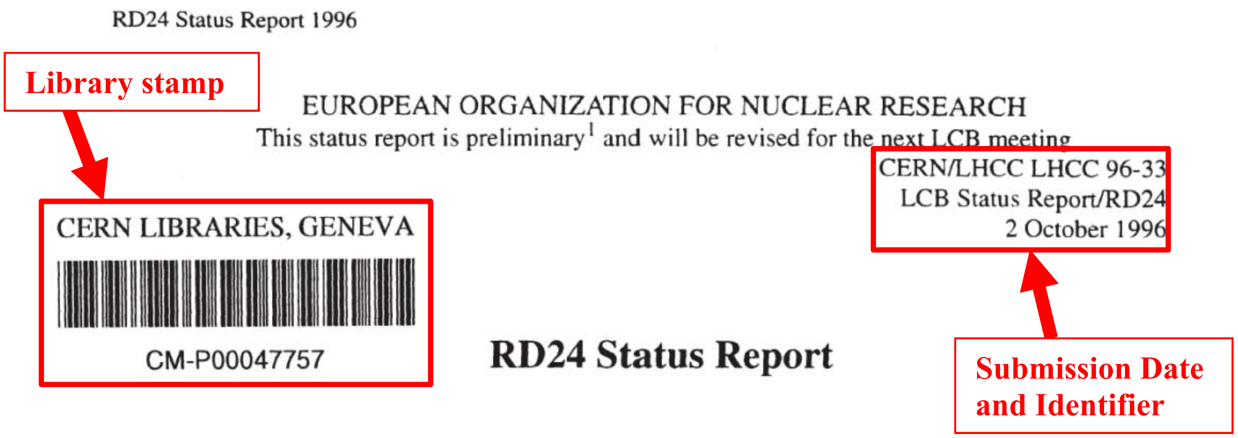
5. Petitioner has not properly established Bogaerts as prior art and, therefore, Ground 2 cannot render Claims 31-33 obvious

To qualify as prior art, Petitioner must show that the reference was publicly available by the critical date. *In re Lister*, 583 F.3d 1307, 1311 (Fed. Cir. 2009). Public accessibility requires that a reference must have been sufficiently accessible to the public interested in the art. *Id*. A reference is considered publicly accessible if it was (1) publicly disseminated or (2) otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could locate it. *Id.* Even where a reference has been stored in a library, this does not mean that a reference is necessarily publicly accessible. *Id.* at 1311-12. Instead, courts have looked to whether the reference was indexed or catalogued by title or subject matter, or was key word searchable. *Id.* at 1312-1317. Petitioner has not provided sufficient evidence under the Federal Rules of Evidence to show that Bogaerts was publicly available prior to the critical date.

Two pieces of information exist on the face of Bogaerts: (1) a CERN library stamp and (2) a date of October 2, 1996 and a reference number CERN/LHCC LHCC 96-33. Petitioner relies on both the stamp and the date to prove the date the reference was available in the CERN library. As set out below, the two pieces of information are not related and are insufficient to prove a date of public availability.

a. The CERN library stamp is undated

The CERN library stamp on the face of the reference is undated. Ex. 1011 at 1. The date on the face of Bogaerts is unrelated to the library stamp. Ex. 2021 at 170-71. Instead, it indicates the date the reference was submitted to the CERN committee for approval by the RD24 project team. *Id.* at ¶ 170-71. Specifically, the header information is shown in Figure I, annotated and excerpted from Ex. 1011. As shown in the Figure I, the date is independent from the library stamp.



RD24 Status Report 1996

Library stamp

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
This status report is preliminary[1] and will be revised for the next LCB meeting

CERN/LHCC LHCC 96-33
LCB Status Report/RD24
2 October 1996

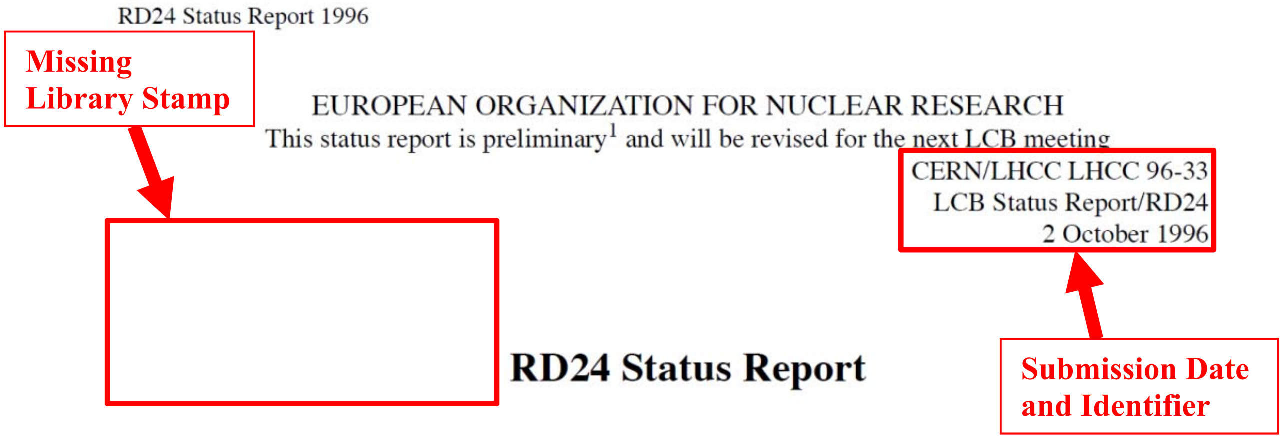
CERN LIBRARIES, GENEVA
CM-P00047757

RD24 Status Report

Submission Date and Identifier

Figure I: Bogaerts Header

The declaration of Mr. Butler from the Internet Archive, submitted by Petitioner as supplemental evidence, confirms no link between the library stamp and the submission date. Ex. 2026. Mr. Butler's declaration included an exhibit that was purportedly a copy of Bogaerts from Internet Archive's Wayback Machine ("the Copy"). Ex. 2026 at 12. The document attached to the declaration does not include the library stamp that the Board relied upon in the institution decision for allowing the reference. Paper No. 11 at 16. The copy does, however, show the submission date and identifier. Ex. 2026 at 12. This proves that the library stamp is unrelated to the October 2, 1996 date and the identifier number. Petitioner has not shown that date on the face of Bogaerts correlates to the date of availability in the CERN library. The heading information from the Copy (Ex. 2026 at 12) missing the library stamp is set out in Figure J below.



RD24 Status Report 1996

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
This status report is preliminary[1] and will be revised for the next LCB meeting

CERN/LHCC LHCC 96-33
LCB Status Report/RD24
2 October 1996

RD24 Status Report

Missing Library Stamp

Submission Date and Identifier

Figure J: Wayback Machine Copy Header (Ex. 2026 at 12)

Additionally, the Copy comes from a server named "sunshine.cern.cd:8080." Ex. 2026 at 10-11. Dr. Lindenstruth, a co-author of the Bogaerts reference, has confirmed that the sunshine server was the personal server of Hans Muller and was not a certified CERN server. Ex. 2021 at ¶ 174. This is evidenced by the non-standard port number, 8080. *Id.* at ¶¶ 174-75. Such a non-standard port is typically used for personal servers not intended to be openly available for public access. *See id.* at ¶ 175. Therefore, the Copy does not prove that Bogaerts was publicly accessible prior to the critical date.

b. The October 2, 1996 date on the face of the Bogaerts reference is not the publication or publicly-available date

Contrary to the Petitioner's claim that the date on the face of the document shows that the Bogaerts reference was published by October 2, 1996, Dr. Lindenstruth, an author of the reference, confirms that October 2, 1996 is merely the date the report was accepted by the CERN/LHCC committee. Ex. 2021 at ¶¶ 170-71. Further, Dr. Lindenstruth is unaware of any certain date prior to the '814 patent's critical date that the report was published, indexed, and made available to the public. *Id.* at ¶ 170. Dr. Lindenstruth believes it was unlikely that the Bogaerts reference was publicly available at the time it was submitted because its content was subject to confidentiality obligations. *Id.*

Petitioner's supplementary evidence to try to prove that the RD24 report was published October 2, 1996 is inadmissible hearsay, and it does not prove that the Bogaerts reference relied upon was published on that date. Fed. R. Evid. 801, 802. Petitioner first points to a declaration from its counsel's research analyst, Mr. Scott, who states that he pulled the document from the CERN library server on April 7, 2015. Ex. 2011. Mr. Scott also states that the bibliographic data states that the document was "submitted by October 2, 1996." *Id.* However, Mr. Scott's guess is insufficient to prove public accessibility. First, Mr. Scott has no personal knowledge of any date on which the reference was actually published. Ex. 2025 at

17. Therefore his testimony is inadmissible under Federal Rule of Evidence 602. Second, the CERN library server simply states that the document was "submitted by October 2, 1996." Ex. 2011 at 439. It does NOT say the document was "published by October 2, 1996." *Id.* The October 2, 1996 date of submission to and acceptance by the committee does not prove that the reference was published or publicly available on that date.

As such, there is not sufficient evidence to proceed on Bogaerts. Petitioner has the burden to prove that the reference is prior art. Petitioner has failed. Therefore, the Bogaerts reference should be removed from consideration as prior art.

V. Conclusion

For the foregoing reasons, Patent Owner respectfully submits that Petitioner has failed to establish by a preponderance of evidence that any of the Instituted Claims are unpatentable under any of 35 U.S.C. §§ 102 or 103. Accordingly, all of the Instituted Claims should be confirmed.

June 11, 2015

Respectfully submitted,
COOLEY LLP

By: ______________________
Wayne O. Stacy
Reg. No. 45,125

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue, N.W
Suite 700
Washington, DC 20004-2400

Tel: (720) 566-4000
Fax: (202) 842-7899

CERTIFICATION OF SERVICE UNDER 37 C.F.R. § 42.6(e)

I, SARA J. BRADFORD, hereby certify that on the 11th of June, 2015 the foregoing **PATENT OWNER'S RESPONSE** was served electronically via email on the following:

Brian Buroker
bburoker@gibsondunn.com

Blair Silver
bsilver@gibsondunn.com

c/o GIBSON, DUNN & CRUTCHER LLP
1050 Connecticut Avenue, N.W.
Washington, D.C. 20036-5306
202.955.8541(telephone)

June 11, 2015
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue, N.W
Suite 700
Washington, DC 20004-2400

Tel: (720) 567-4000
Fax: (202) 842-7899

Respectfully submitted,
COOLEY LLP

By: /Sara J. Bradford/
Sara J. Bradford
Reg. No. 68,715